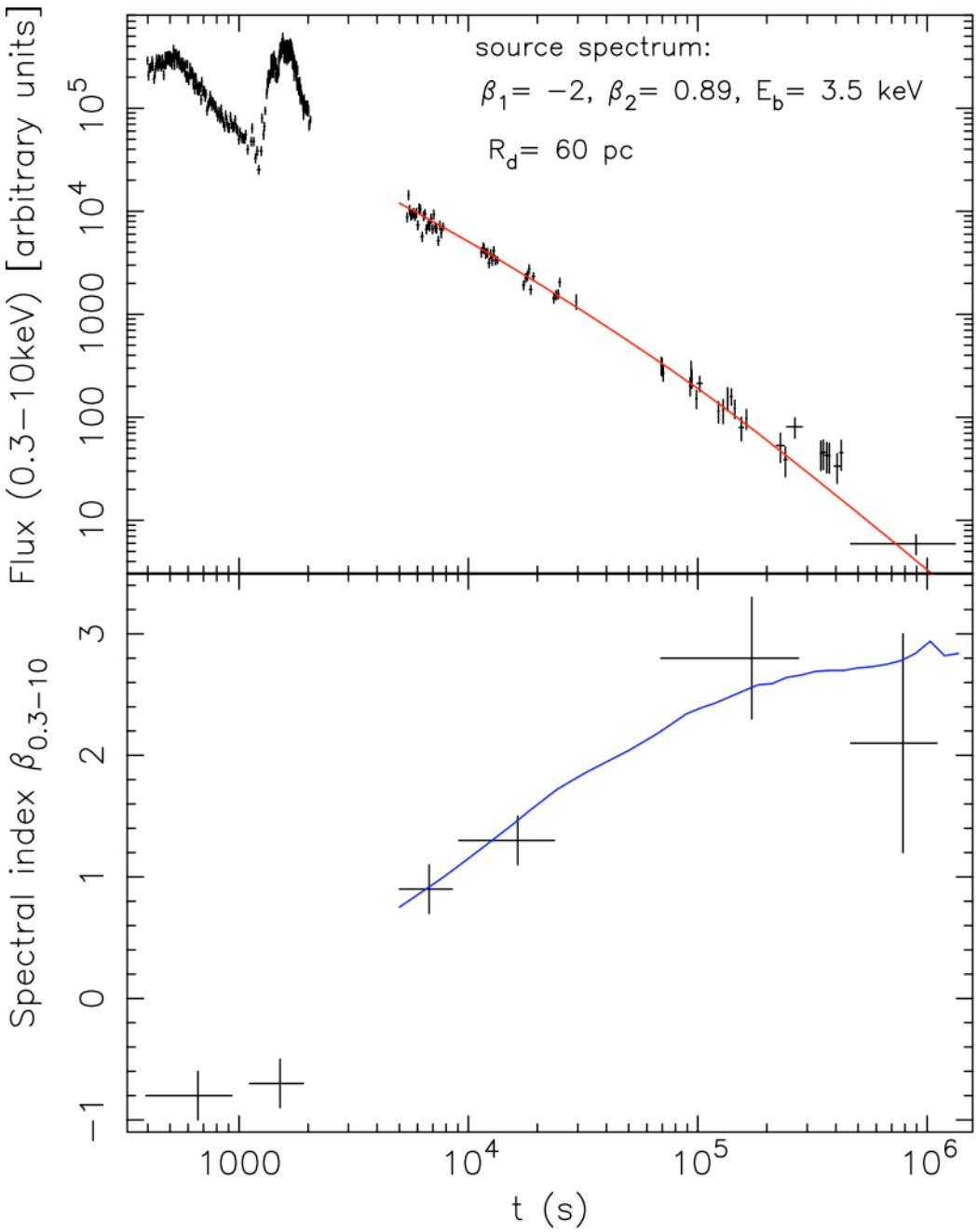
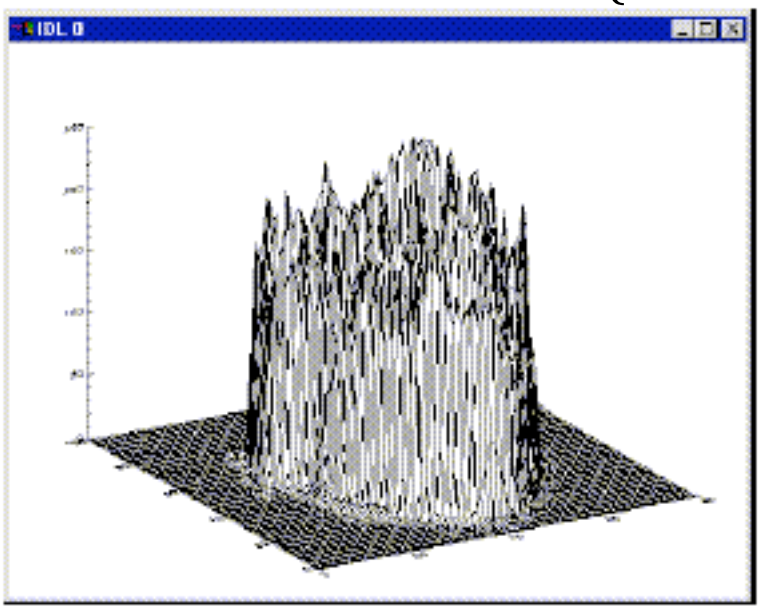
LECTURE 8

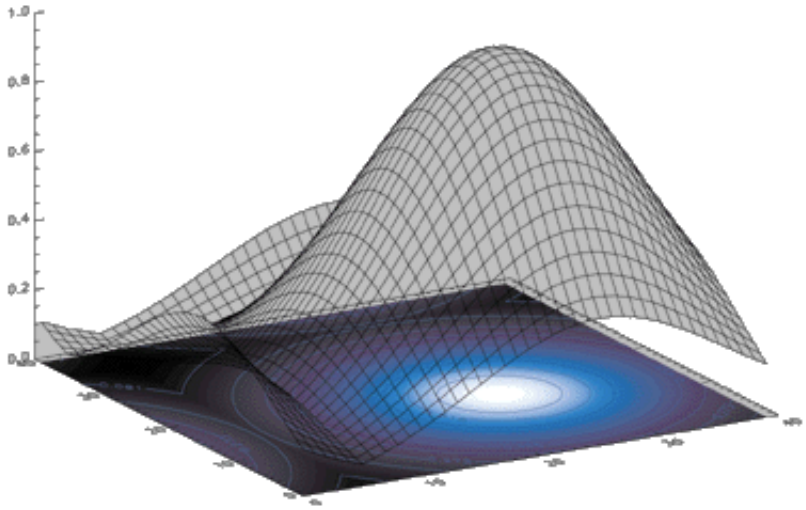
Data Analysis II: Plotting



IDL (Interactive Data Language)

- Commercial Software
- Widely used in sciences
- Similar syntax to Fortran and C
- Becoming a *de facto* standard in astronomy
- Free version available (GNU Data Language)





Pros

- Simple and flexible syntax
- Can read most data formats, including FITS
- Interactive
- Arbitrarily complex plots. If you can dream it up IDL can plot it.
- Powerful data manipulation language
    - Fitting functions
    - Statistics
    - Time series analysis
    - Arbitrary data manipulation
- Complete programming language
- Powerful built-in debugger
- Can create your own procedures and libraries
- Many free libraries exist
- IDL can be called from inside software
- Platform independent
- Operate on entire *N*-dimensional arrays.
- Allow high-level abstraction of data

Cons

- Expensive! (but GDL is free)
- Steep learning curve (no-one knows all of IDL)
- Plots to the screen can look different from plots to a file.
- Not optimized for astronomy journals

Numbers and Arrays

Variable do not need to be declared. Type is determined from context.
Type “help, <variable name>” to get info on that variable.

```
IDL> a = 2
IDL> b = 5.0
IDL> c = a + b
IDL> help, a, b, c
A               INT       =        2
B               FLOAT     =      5.00000
C               FLOAT     =      7.00000
```

Create an $m \times n$ array.
Array dimensions are columns × rows.
“print, <variable name>” prints that variable.

```
IDL> array = indgen (5, 5)
IDL> help, array.
ARRAY           INT       = Array[5, 5]
IDL> print, array
      0       1       2       3       4
      5       6       7       8       9
     10      11      12      13      14
     15      16      17      18      19
     20      21      22      23      24
```

Operate on an entire array at once.

```
IDL> array = array * a
IDL> print, array
      0       2       4       6       8
     10      12      14      16      18
     20      22      24      26      28
     30      32      34      36      38
     40      42      44      46      48
```

Access an element of an array.
Array indices start at zero *not one*.

```
IDL> print, array [1,3]
      32
```

Operate on elements of an array.

```
IDL> test = array[1,3]*10
IDL> print, test
     320
```

Example of *X*-ray light curve data:

Date [days]	L_x [cgs]	L_x_err [cgs]
8.7550	0.3296	0.13
11.465	0.4525	0.17
14.275	0.8877	0.23
16.785	1.2492	0.26
20.076	1.4000	0.26
25.280	1.5023	0.24
29.220	1.4210	0.22
33.681	1.4325	0.27
38.459	1.1971	0.24
43.128	1.1777	0.27
47.208	0.9340	0.20
51.530	0.7605	0.21
61.873	0.5783	0.18

Example of an IDL program to plot *X*-ray light curve data:

```
; Give the program a name:

pro lightcurve

; Write output to a PostScript file:

file = 'lightcurve'
filename = file + '.ps'
set plot, 'ps'
device, file=filename

; Read in data:

header = STRARR(2)  ; array of strings
array = FLTARR(3,13)  ; array of floating point numbers
OPENR, SN2008bo, 'lightcurve.dat', /GET_LUN
READF, SN2008bo, header, array
date = REFORM(array[0,*])
Lx = REFORM(array[1,*])
Lx_err = REFORM(array[2,*])

; Plot axis and data:

USERSYM, [0.8,-0.8,-0.8,0.8,0.8], [0.8,0.8,-0.8,-0.8,0.8], /FILL
PLOT, date, Lx, XRANGE=[0,70], YRANGE=[0.01,2.0], $
XTITLE='Date', $
YTITLE='Luminosity', CHARSIZE=1.5

; Close Postscript file:

device, /close
set_plot, 'x'

END
```

Running IDL

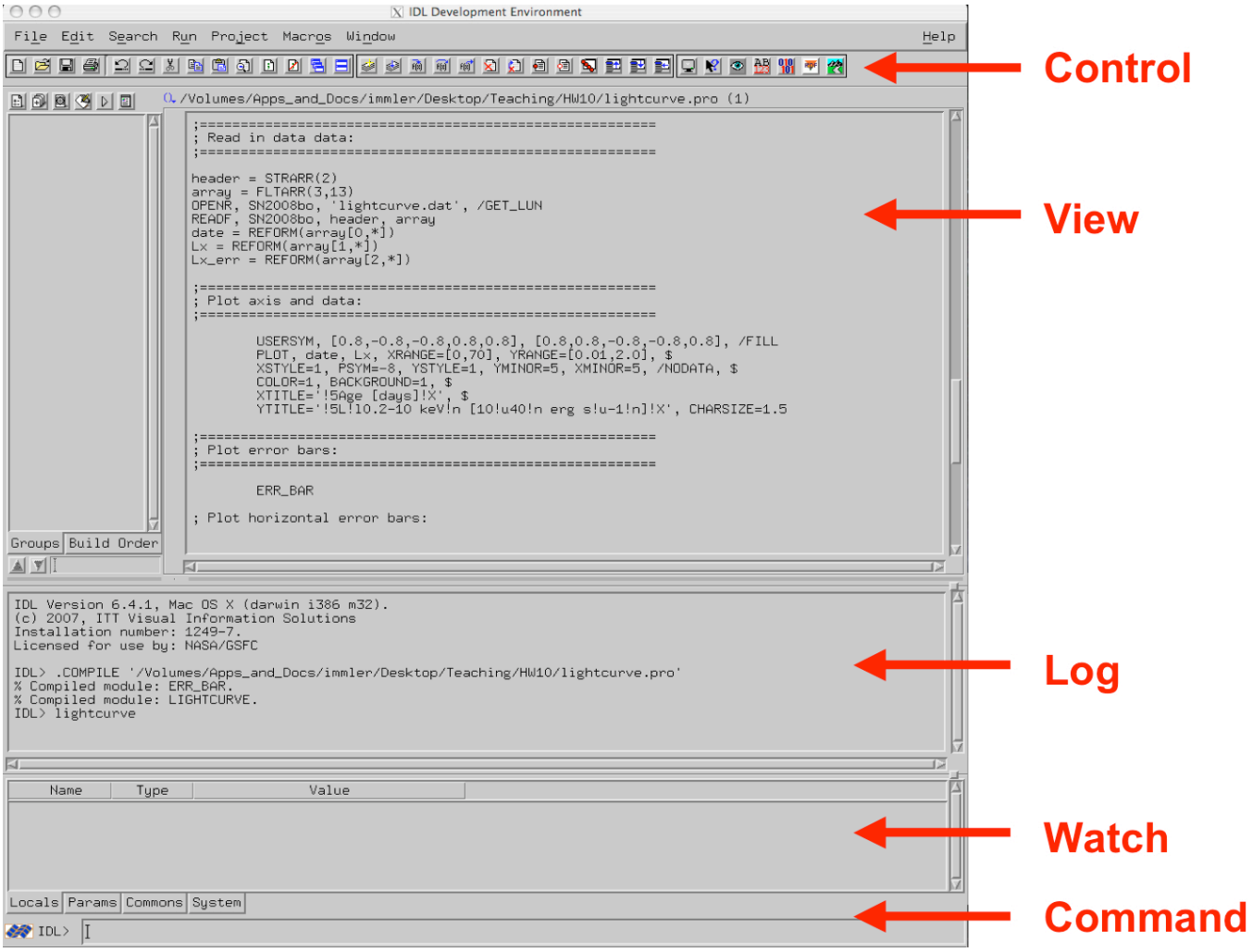
Command Line

```
% idl
IDL Version 6.4.1, Mac OS X (darwin i386 m32). (c) 2007, ITT Visual
Information Solutions
Installation number: 1249-7.
Licensed for use by: NASA/GSFC

IDL>
```

GUI

```
% idled
```



Getting Help

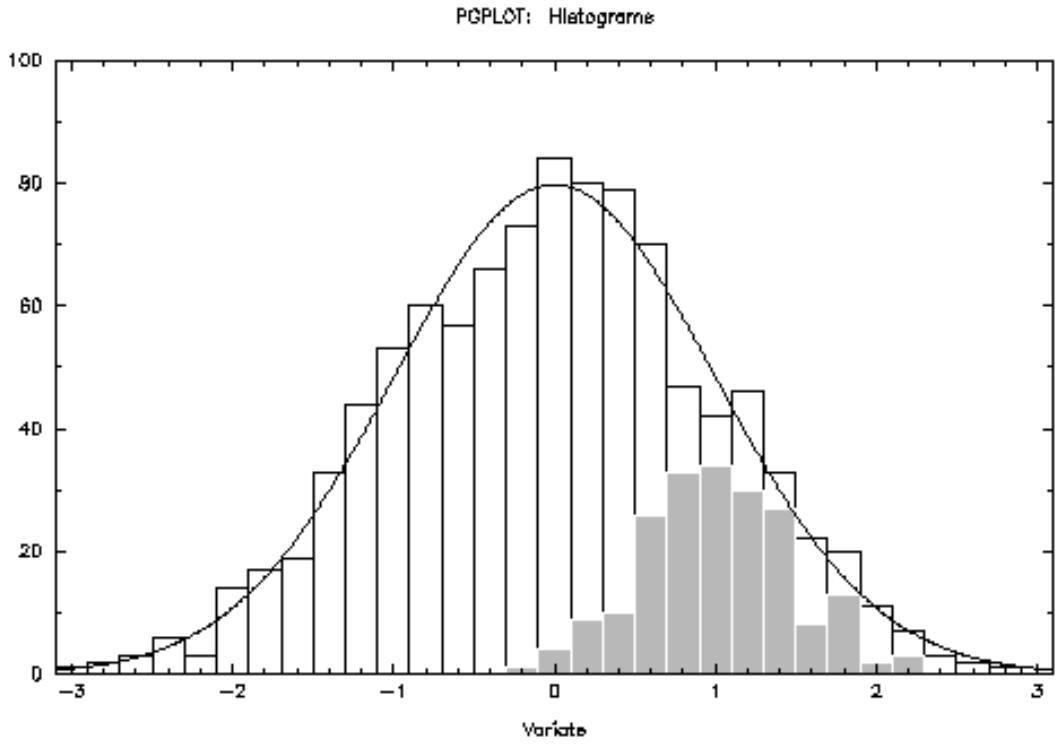
To get help type “?” at an IDL prompt.

```
IDL> ?
```

The details of how help is displayed depends on how IDL was set up on the system that you are using.

PGPlot

- Graphics subroutine library
- Primarily used by programmers
- Written in Fortran 77
- Device independent
- Widely used in astronomical software
- Very powerful
- Free



SM (SuperMongo)

- Interactive plotting package
- Libraries can be called by software
- Similar to IDL, but not as powerful
- Easier to learn than IDL
- Can do data analysis
    - Fourier transforms
    - Statistics
    - Function fitting
- Simple programming language included.
- Widely used in astronomy
- Being supplanted by IDL
- Plot are naturally formatted for astronomy journals

- ***Do not confuse it with an older (and still widely used) plotting package called "mongo"!***